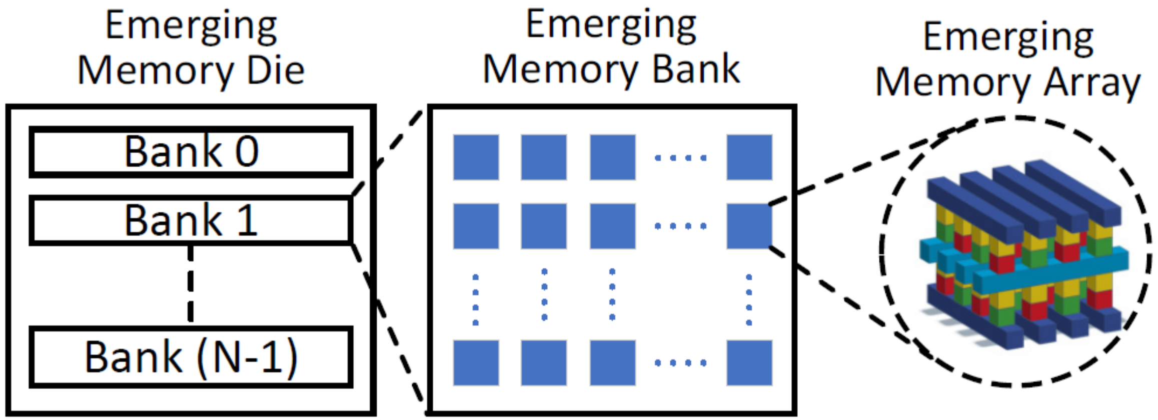
ECC Replay: A Practical and Efficient Scheme To Tolerate High Stuck Bit Rate For Emerging Memories

Lunkai Zhang, Nate Franklin
Western Digital

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Emerging Memories



- Emerging Memories such as PCM, ReRAM, MRAM and Selector-Only Memory, offer unique value propositions, such as higher density, persistency or radiation hardness.
- Typical Emerging Memory Die Architecture:
  - Each die consists of multiple banks.
  - Each bank consists of many crossbar arrays.
  - In a crossbar array, memory cells are sandwiched between bitlines and wordlines.
  - In a memory read/write, each array contributes a single bit.

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Motivation: High Stuck Bit Rate In Emerging Memories

- Problem Statement: Stuck failure is usually the major failure type in end-of-life emerging memory media.
- Opportunity: Stuck failures can provide extra information for data recovering.
- Challenge: How to design a high efficiency, low complexity scheme to tolerate high-stuck bit rate with minimal changes to existing memory controller architecture?

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Background: Uncorrectable and Miscorrections

Any practical emerging memory system is protected by ECC (Error Correction Codes), and any ECC system is subject to two types of failures.

- **Uncorrectables**. If the ECC decoder cannot provide a valid ECC outcome, there is an uncorrectable.
  - Uncorrectables are **detectable** from ECC outcome.
- **Miscorrections**. If the decoder provides a valid ECC outcome but it is different from the original data, there is a miscorrection.
  - Miscorrections are **undetectable** from ECC outcome.

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Background: Memory Reliability Metrics

- BER (Bit Error Rate):

$$BER = \frac{\#Bit\_Errors}{\#Total\_Read\_Bits}$$

- Fail Rate:

$$Fail\_Rate = \frac{\#Uncorrectable\_Codewords}{\#Total\_Read\_Codewords}$$

- UBER (Uncorrectable Bit Error Rate): → Reliability Metric for Uncorrectables

$$UBER = \frac{\#Uncorrectable\_Codewords}{\#Total\_Read\_Bits}$$

- MISC Rate (Miscorrection Rate): → Reliability Metric for Miscorrections

$$MISC\_Rate = \frac{\#Miscorrected\_Codewords}{\#Total\_Read\_Codewords}$$

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Baseline ECC Configuration

- Without loss of generality, we use BCH-6 ECC:
  - [n, k, d] representation [572, 512, 13].
  - 572 total bits, 512 user bits, 60 parity bits.
  - Correcting up to 6 bit errors.

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Binomial Failure Distribution

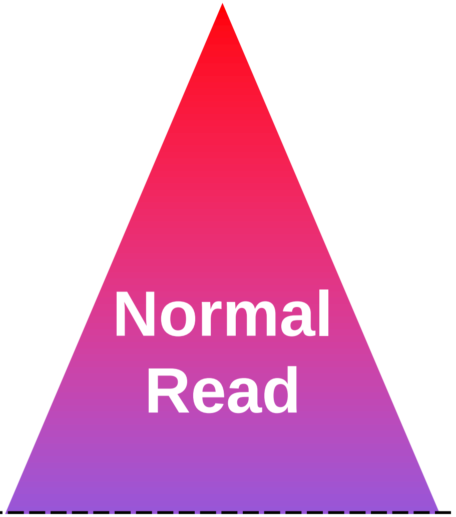
- In this work, we assume bit fails of emerging memory are following binomial distribution.

© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Let’s Start With A Normal Read with Baseline BER

Consider we have 1E-5 BER (of any kind).
We need to achieve an UBER target of 1E-18, and MISC Rate target of 1E-22.

Every codeword read is a Normal Read with BCH-6 ECC.



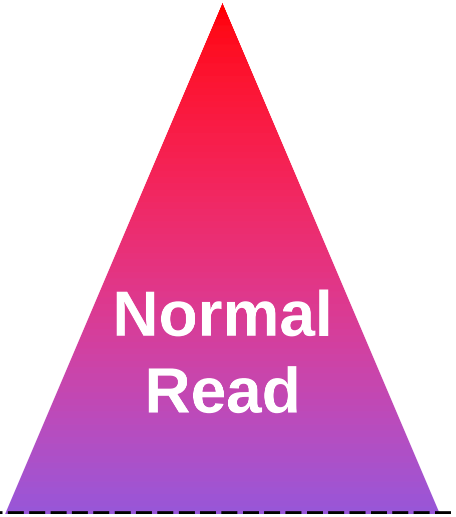
Fail Rate Requirement (As low as possible)	**UBER Requirement (Target 1E-18)**	**MISC Rate Requirement (Target 1E-22)**
3.8E-20	6.7E-23 (**Meet Target**)	1.6E-24 (**Meet Target**)

© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Now We Would Like to Tolerate More Stuck Bits …

Consider we have 1E-5 Soft BER **and 3E-4 Stuck Bit Rate**.

Every codeword read is a Normal Read with BCH-6 ECC.



Fail Rate Requirement (As low as possible)	UBER Requirement (Target 1E-18)	MISC Rate Requirement (Target 1E-22)
9.7E-12	1.7E-14 (**Miss Target**)	4.0E-16 (**Miss Target**)

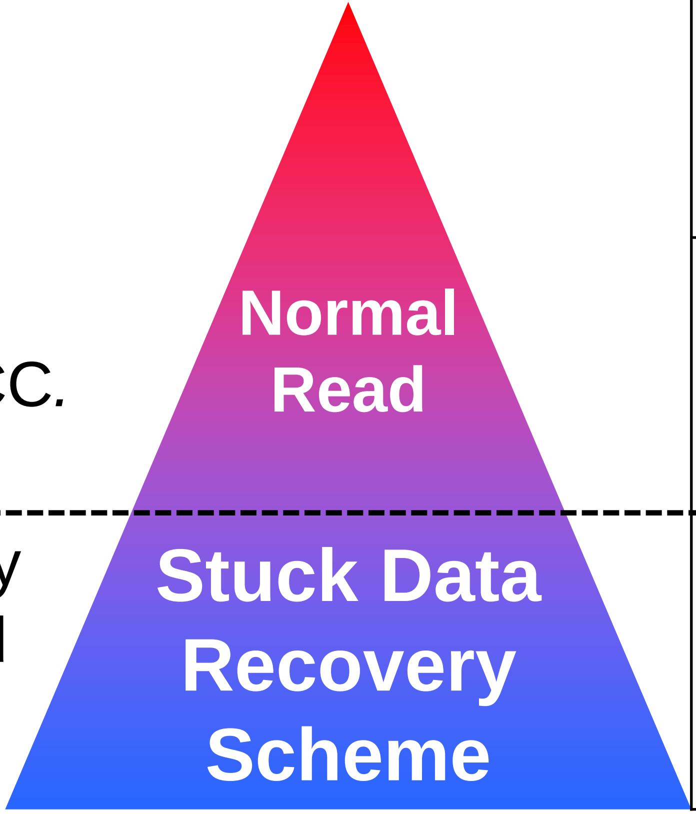
© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

How About A Hierarchical ECC Framework?

Consider we have 1E-5 Soft BER **and 3E-4 Stuck Bit Rate**.

Every codeword read starts with a Normal Read using BCH-6 ECC.

A Stuck Data Recovery Scheme gets triggered only when the *Normal Read* fails.



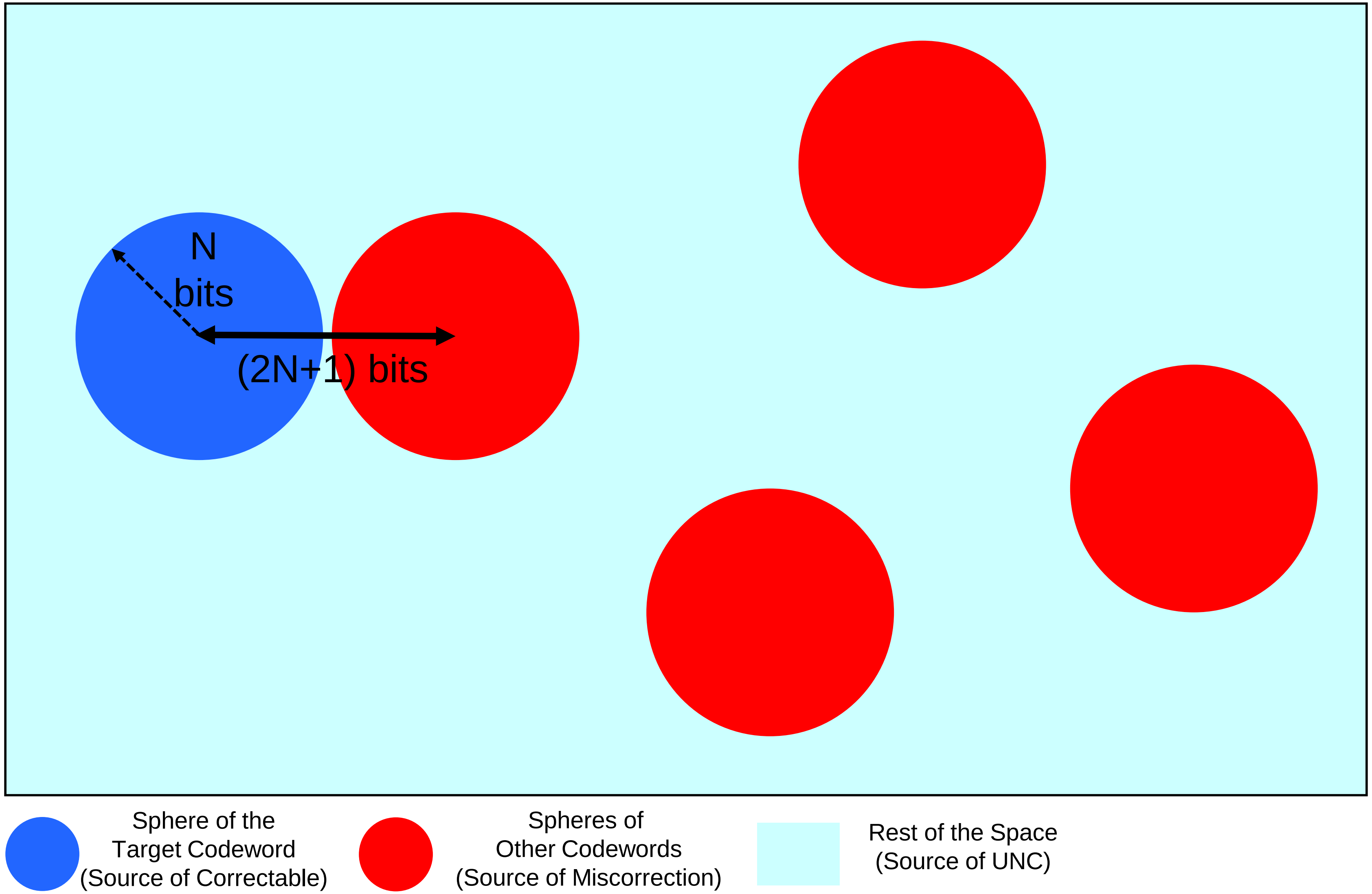
Fail Rate Requirement (As low as possible)	UBER Requirement (Target 1E-18)	MISC Rate Requirement (Target 1E-22)
9.7E-12	NA (Uncorrectable will be fixed in next layer)	4.0E-16 (**Miss Target**)
		???

Before we introduce the stuck data recovery scheme, we need to first solve the high MISC Rate issue of Normal Read layer.

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Why Miscorrections Happen In BCH ECC?

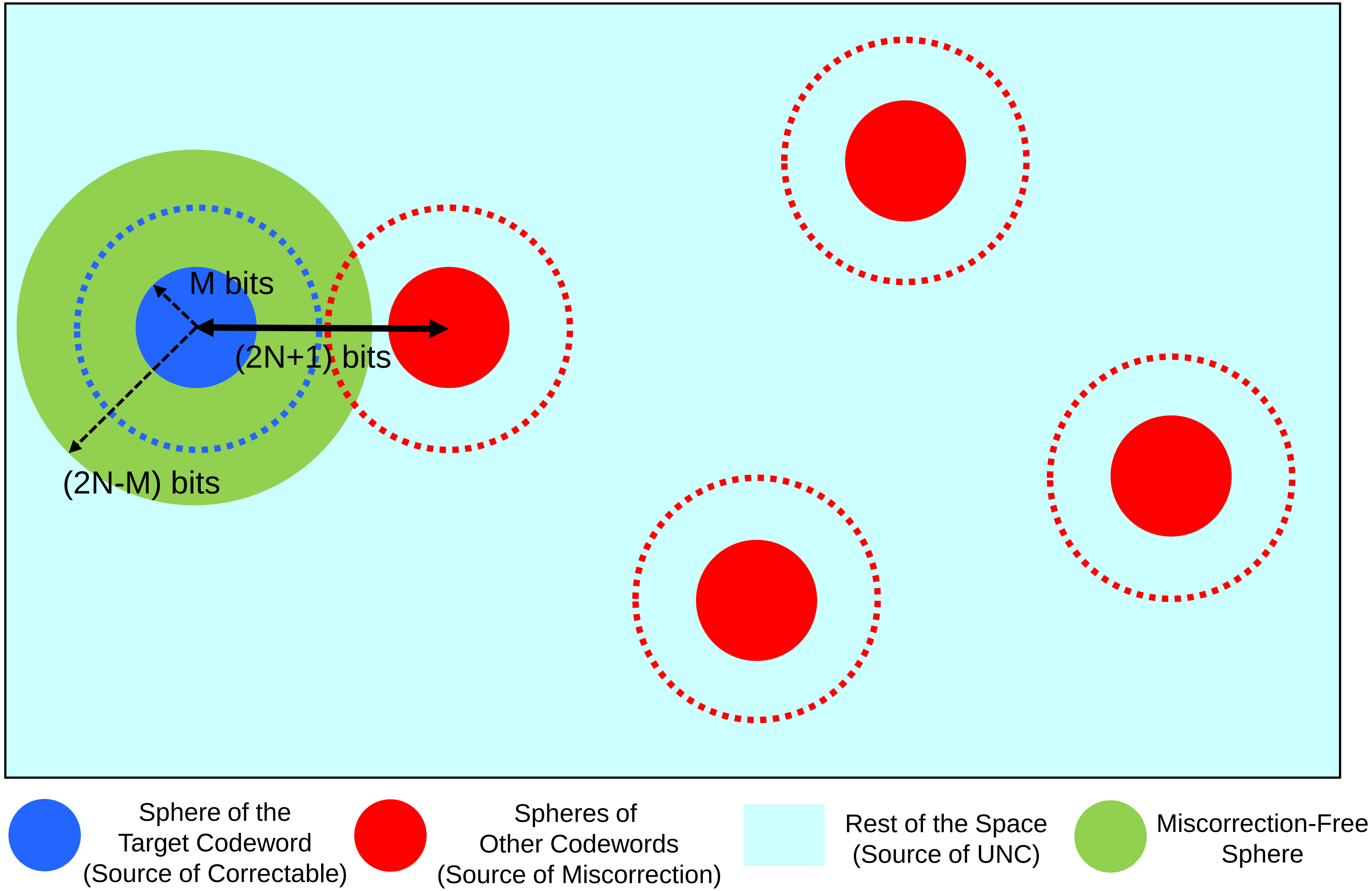
- If BCH code correcting up to N bits. When we have equal or less than N bit errors, the codeword is correctable.
- There are many other codewords, if the ECC outcome falls into these codewords, we have a **miscorrection**. The closest bit difference (Hamming Distance) in any two codewords is 2N+1.
- The rest of the space is **uncorrectable** without miscorrection.



 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Reducing Miscorrection Probability in BCH ECC Through Under-Correction

- We can reduce miscorrection probability by under-correcting. I.e., only trust the ECC result with up-to M bit corrections, where M is smaller than N.
- Why Under-correction reduces miscorrection?
  - If we under-correct in BCH, the correctable combination of all other codewords is reduced. As a result, the probability of falling into other codewords' sphere (i.e., miscorrection) is reduced.
  - If we under-correct in BCH, the space with no miscorrection also grows. For example, if we only correct up to 3 bits in BCH-6, we won't have miscorrection with up to 9 (2*6-3) bit errors.



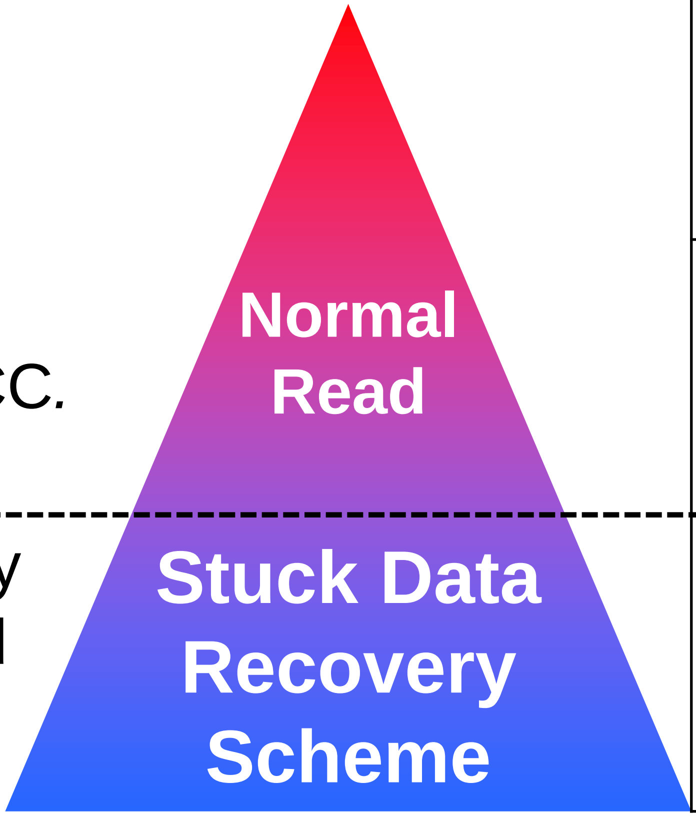
© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Fixing MISC Rate Issue In Normal Read Layer With Under-Correction

Consider we have 1E-5 Soft BER **and 3E-4 Stuck Bit Rate**.

Every codeword read starts with a Normal Read using BCH-6 ECC.

A Stuck Data Recovery Scheme gets triggered only when the *Normal Read* fails.



Fail Rate Requirement (As low as possible)	UBER Requirement (Target 1E-18)	MISC Rate Requirement (Target 1E-22)
9.7E-12	NA (Uncorrectable will be fixed in next layer)	4.0E-16 (**Miss Target**)
???	???	???

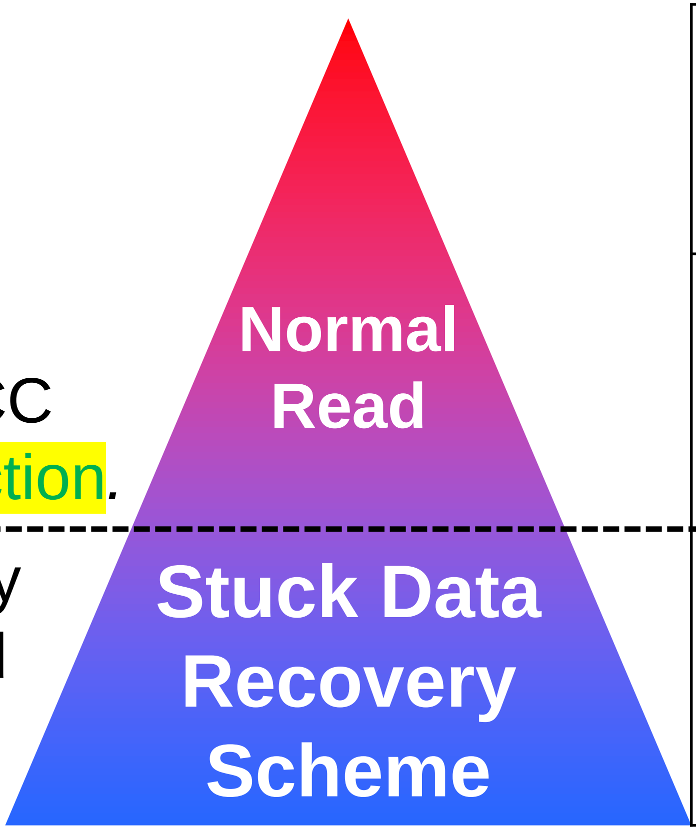
© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Fixing MISC Rate Issue In Normal Read Layer With Under-Correction

Consider we have 1E-5 Soft BER **and 3E-4 Stuck Bit Rate**.

Every codeword read starts with a Normal Read using BCH-6 ECC with 3-bit under-correction.

A Stuck Data Recovery Scheme gets triggered only when the *Normal Read* fails.



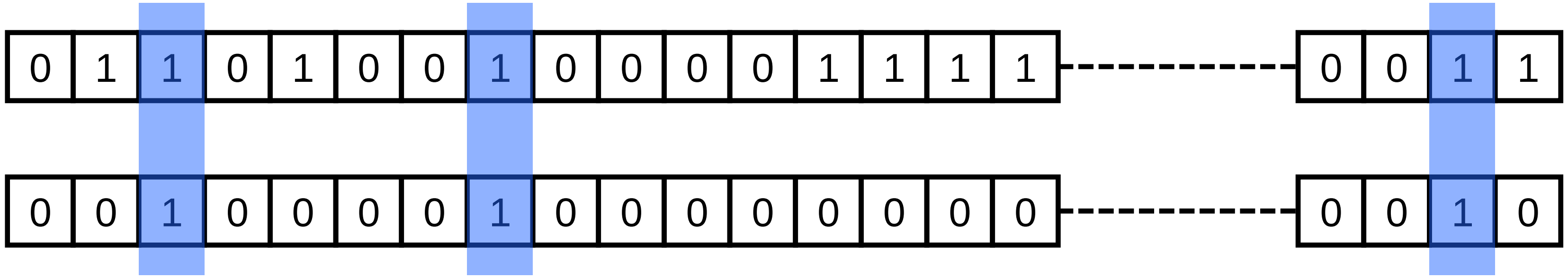
Fail Rate Requirement (As low as possible)	UBER Requirement (Target 1E-18)	MISC Rate Requirement (Target 1E-22)
2.7E-6 (Low enough trigger rate for next layer)	NA (Uncorrectable will be fixed in next layer)	2.6E-28 (**Meet Target**)
???	???	???

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

ECC Replay: the Stuck Data Recovery Scheme

Step 1: Raw Uncorrectable Data

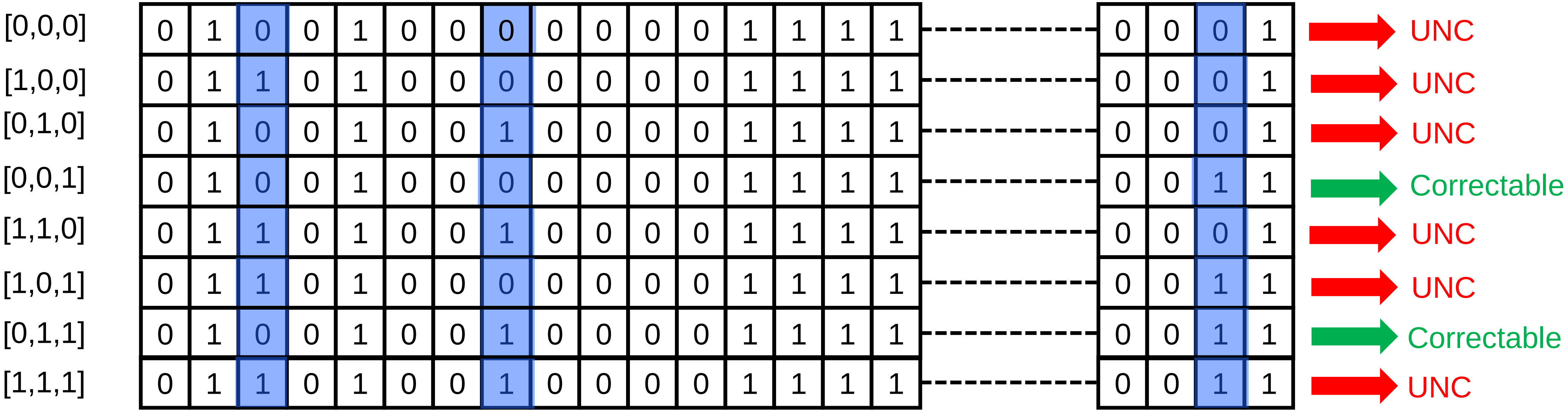
Step 2: Write All-0
then Read Back to detect stuck bits.



Step 3: Get all the bit-flip combinations of the stuck bits: [0,0,0],[1,0,0],[0,1,0],[0,0,1],[1,1,0],[1,0,1],[0,1,1],[1,1,1]

Stuck Bits

Step 4: For each combination, replace the stuck-at bits in raw data with the combination and go through ECC again:



Step 5. ECC Result Interpretation (Next Slide).

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

ECC Replay: the Stuck Data Recovery Scheme (Cont.)

Step 5. ECC Result Interpretation:

- Case #1: One or multiple combinations are correctable, and the ECC outcomes of these combinations are the same.

=> The shared ECC outcome of all the correctable combinations is returned as corrected data.

- Case #2: Multiple combinations are correctable, and there are more than one ECC outcomes. In this case, we don't know which outcome(s) are miscorrected.

=> Return uncorrectable to avoid miscorrection.

- Case #3: No combination is correctable.

=> Return uncorrectable.

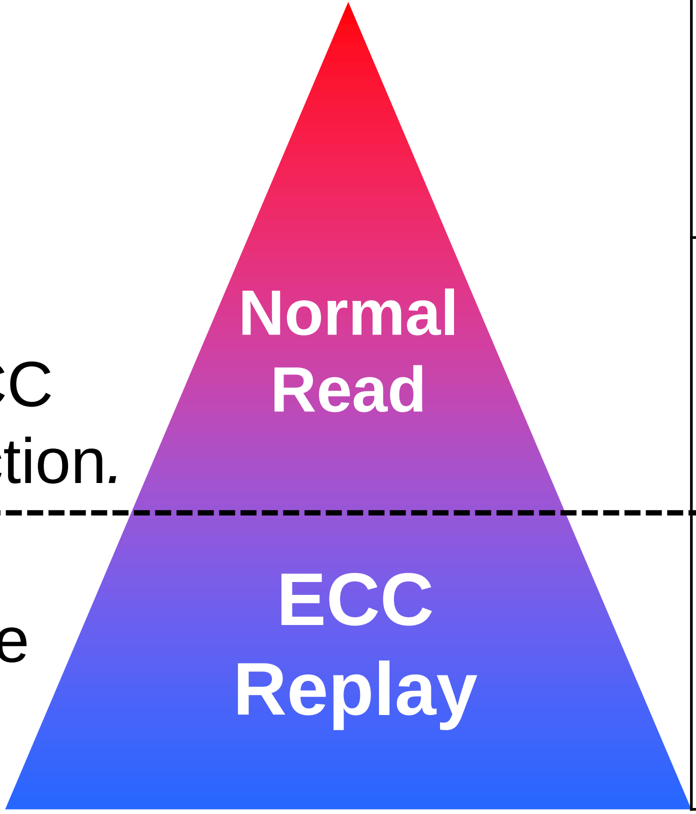
© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Effectiveness of ECC Replay

Consider we have 1E-5 Soft BER **and 3E-4 Stuck Bit Rate**.

Every codeword read starts with a Normal Read using BCH-6 ECC with 3-bit under-correction.

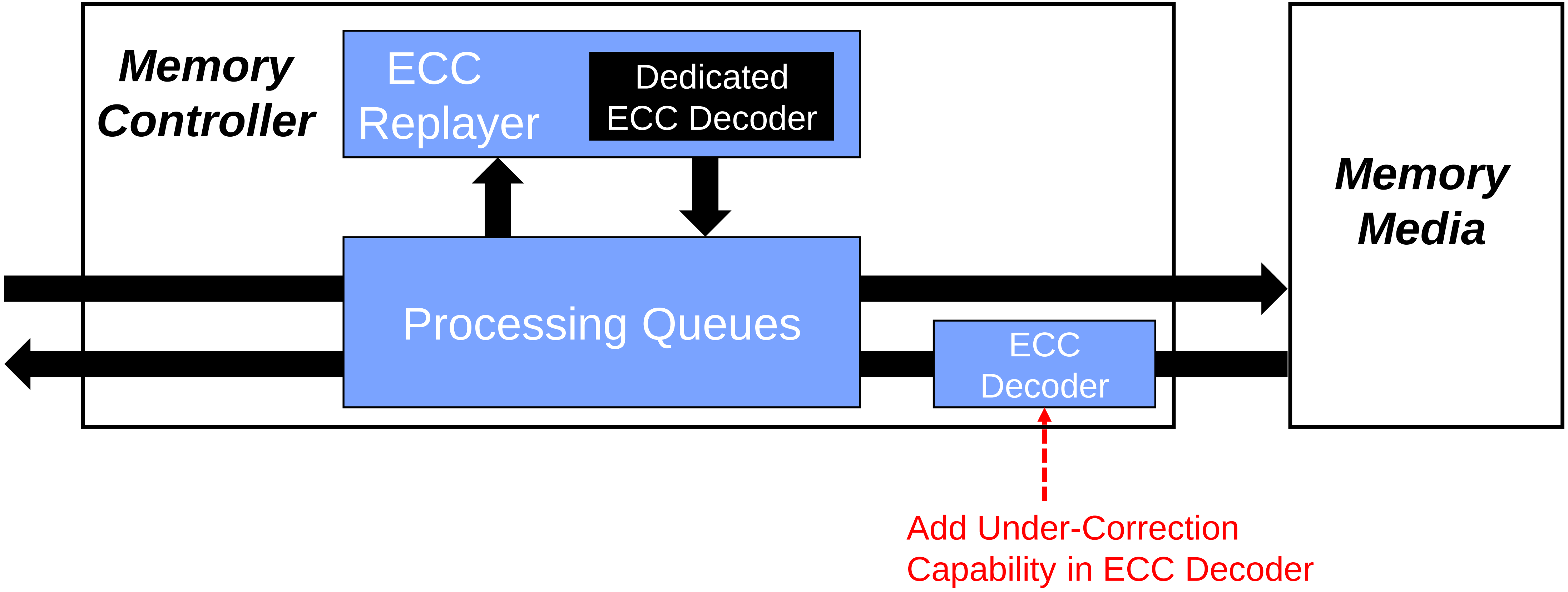
ECC Replay gets triggered only when the *Normal Read* fails.



	Fail Rate Requirement (As low as possible)	UBER Requirement (Target 1E-18)	MISC Rate Requirement (Target 1E-22)
Normal Read	2.7E-6 (Low enough trigger rate for next layer)	NA (Uncorrectables will be fixed in ECC Replay)	2.6E-28 (**Meet Target**)
ECC Replay	Not Important	6.9E-19 (**Meet Target**)	1.6E-24 (**Meet Target**)

© 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

How to Integrate ECC Replay Into A Typical Memory Controller?



 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Performance Evaluation

- In-House Cycle-Accurate Emerging Memory Simulator.
  - 1E-5 Soft BER and 3E-4 Stuck Bit Rate.
- Five Typical Memory Qualification Workloads:
  - Rand All-Rd and Rand 2R1W;
  - Seq All-Rd and Seq 2R1W;
  - JEDEC-like;
- For all five benchmarks, performance impact of ECC Replay is negligible:
  - Both average latency of bandwidth loss are within 0.2%;
  - ECC Replay doesn’t introduce extra tail latency for read request;
  - This is because ECC Replay has a really low trigger rate in 1E-6 range.

 © 2025 WESTERN DIGITAL CORPORATION OR ITS AFFILIATES ALL RIGHTS RESERVED

Summary

We introduce ECC Replay, an efficient and practical scheme to tolerate high stuck bit rate in emerging memories.

- It improves stuck bit rate tolerance from 1E-5 to 3E-4, which is 30x improvement;
- It has negligible performance impact;
- It is easy to be integrated into existing memory controller architecture.

TM